PRACTICA SPRING

Introducción

El **Spring Framework** (también conocido simplemente como **Spring**) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java. La primera versión fue escrita por Rod Jonhson, quien lo lanzó primero con la publicación de su libro *Expert One-on-One Java EE Design and Development* (Wrox Press, octubre 2002).

El framework fue lanzado inicialmente bajo Apache 2.0 License en junio de 2003. El primer gran lanzamiento hito fue la versión 1.0, que apareció en marzo de 2004 y fue seguida por otros hitos en septiembre de 2004 y marzo de 2005.

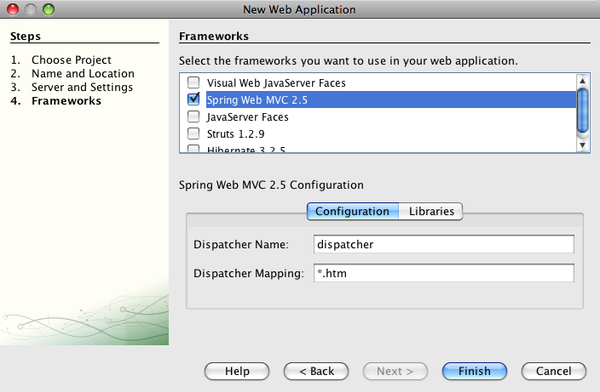
A pesar de que Spring Framework no obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al ser considerado como una alternativa y sustituto del modelo de Enterprise JavaBeans. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria.

Mientras que las características fundamentales de este framework pueden emplearse en cualquier aplicación hecha en Java, existen muchas extensiones y mejoras para construir aplicaciones basadas en web por encima de la plataforma empresarial de Java (Java Enterprise Plataform).

A partir de 2009 las actualizaciones del producto (en su forma binaria) estarán disponibles únicamente para la última versión publicada del Framework. Para acceder a las actualizaciones en forma binaria para versiones anteriores habrá que pagar una subscripción. Sin embargo, estas actualizaciones estarán disponibles libremente (y gratuitamente) en forma de código fuente en los repositorios públicos del proyecto

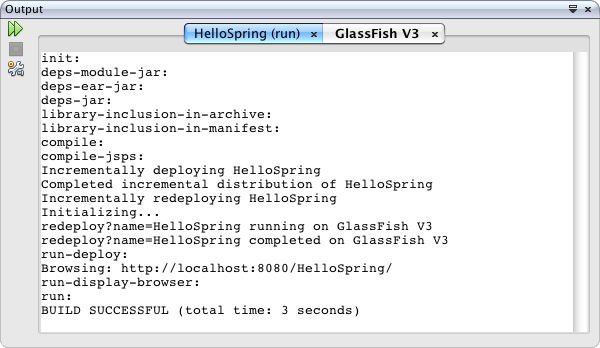
Creación de una web MVC con Spring

* Elija Nuevo Proyecto (Ctrl + Mayúsculas + N) en el menú Archivo del IDE. En categorías selecciona Java Web. (Si usted está utilizando NetBeans 6.1, seleccione Web.) En Proyectos, seleccione Aplicación Web. Haga clic en Siguiente.
* En Nombre del proyecto, el tipo de HelloSpring . Haga clic en Siguiente.
* En el paso 3, el servidor y la configuración, seleccione el servidor que va a trabajar con el servidor de la lista desplegable. Agregar el resto de ajustes en sus valores predeterminados y haga clic en Siguiente.
* En el paso 4, el panel Marcos, seleccione Spring Web MVC 2.5:



Antes de hacer cualquier cambio a los archivos de proyecto, intente ejecutar el nuevo proyecto en el IDE:

* En la ventana de Proyectos, haga clic en el nodo del proyecto y seleccione Ejecutar proyecto. El IDE se inicia automáticamente el servidor si no se está ejecutando, se basa entonces implementa la aplicación a la misma. La aplicación se ejecuta utilizando los datos de configuración que figuran en el dispatcher-servlet.xml . Tome nota de cualquier resultado que aparece en la ventana de salida del IDE (Ventana> Salida). La salida generada se completa con una BUILD SUCCESSFUL mensaje:



En el descriptor de despliegue, tenga en cuenta que todas las solicitudes de los patrones de URL que coincidan con \*.htm se asignan a la primavera de DispatcherServlet :

*<servlet>*

*<servlet-name> despachador </ servlet-name>*

*<servlet-class> org.springframework.web.servlet.DispatcherServlet </ servlet-class>*

*<load-on-startup> 2 </ load-on-startup>*

*</ Servlet>*

*<servlet-mapping>*

*<servlet-name> despachador </ servlet-name>*

*<url-pattern> \*. htm </ url-pattern>*

*</ Servlet-mapping>*

El DispatcherServlet maneja las solicitudes entrantes en función de los valores de configuración se encuentran en dispatcher-servlet.xml . Abierto dispatcher-servlet.xml haciendo clic en su ficha en el editor de código. Tenga en cuenta el siguiente código:

*<Bean id = "UrlMapping" class = "org.springframework.web.servlet.handler. SimpleUrlHandlerMapping ">*

*<propiedad name="mappings">*

*<props>*

*<prop key="/index.htm"> IndexController </ prop>*

*</ Props>*

*</ Propiedad>*

*</ Bean>*

*<Bean id = "viewResolver"*

*class = "org.springframework.web.servlet.view. InternalResourceViewResolver "*

*t: prefix = "/ WEB-INF/jsp /"*

*p: = sufijo ". jsp" />*

*<Nombre del bean = "IndexController"*

*class = "org.springframework.web.servlet.mvc. ParameterizableViewController "*

*t: viewName = "Índice" />*

Comience por la creación de la HelloService clase.

* En la ventana de Proyectos, haga clic en el nodo del proyecto y elija New> Java Class.
* En el Asistente para nueva clase Java que muestra, entre HelloService de Nombre de clase, y el tipo de service para el nombre del paquete para crear un nuevo paquete para la clase. Haga clic en Finalizar. El IDE crea la nueva clase y lo abre en el Editor de código fuente. También tenga en cuenta que la ventana de proyectos se actualizará automáticamente para reflejar los cambios.

El HelloService clase realiza un servicio muy simple. Se necesita un nombre como un parámetro, y se prepara y devuelve una String que incluye el nombre. En el Editor de código fuente, cree la siguiente sayHello() método de la clase.

*HelloService public class {*

*public String sayHello (String nombre) {*

*return "Hola" + nombre + "";*

*}*

*}*

*public HelloController () {*

[*setCommandClass*](http://translate.googleusercontent.com/translate_c?hl=es&ie=UTF8&prev=_t&rurl=translate.google.com.mx&sl=en&tl=es&u=http://static.springframework.org/spring/docs/2.5.x/api/org/springframework/web/servlet/mvc/BaseCommandController.html&usg=ALkJrhiY4oUaQa7c1pz9rDSDprXhzFz6NA#setCommandClass%28java.lang.Class%29) *(Nombre de la clase.);*

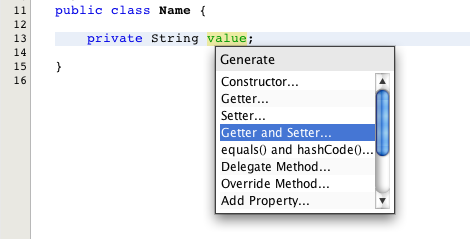
[*setCommandName*](http://translate.googleusercontent.com/translate_c?hl=es&ie=UTF8&prev=_t&rurl=translate.google.com.mx&sl=en&tl=es&u=http://static.springframework.org/spring/docs/2.5.x/api/org/springframework/web/servlet/mvc/BaseCommandController.html&usg=ALkJrhiY4oUaQa7c1pz9rDSDprXhzFz6NA#setCommandName%28java.lang.String%29) *("nombre");*

[*setSuccessView*](http://translate.googleusercontent.com/translate_c?hl=es&ie=UTF8&prev=_t&rurl=translate.google.com.mx&sl=en&tl=es&u=http://static.springframework.org/spring/docs/2.5.x/api/org/springframework/web/servlet/mvc/SimpleFormController.html&usg=ALkJrhhQozm0G5pj-ftVgjyhEP1OL9oClQ#setSuccessView%28java.lang.String%29) *("hola Ver");*

[*setFormView*](http://translate.googleusercontent.com/translate_c?hl=es&ie=UTF8&prev=_t&rurl=translate.google.com.mx&sl=en&tl=es&u=http://static.springframework.org/spring/docs/2.5.x/api/org/springframework/web/servlet/mvc/SimpleFormController.html&usg=ALkJrhhQozm0G5pj-ftVgjyhEP1OL9oClQ#setFormView%28java.lang.String%29) *("Ver nombre");*

*}*

Valor de la cadena



En el cuadro de diálogo que aparece, seleccione el value : String opción, haga clic en Aceptar. El getValue() y setValue() métodos se agregan a la Name clase:

*getValue public String () {*

*valor de retorno;*

*}*

*setValue public void (String value) {*

*This.Value = valor;*

}

La declaración de importación se añade lo siguiente a la parte superior del archivo:

*importar* [*org.springframework.web.servlet.ModelAndView*](http://translate.googleusercontent.com/translate_c?hl=es&ie=UTF8&prev=_t&rurl=translate.google.com.mx&sl=en&tl=es&u=http://static.springsource.org/spring/docs/2.5.x/api/org/springframework/web/servlet/ModelAndView.html&usg=ALkJrhi76U0GAI8z02fugYlBKKbGyqFD_w)

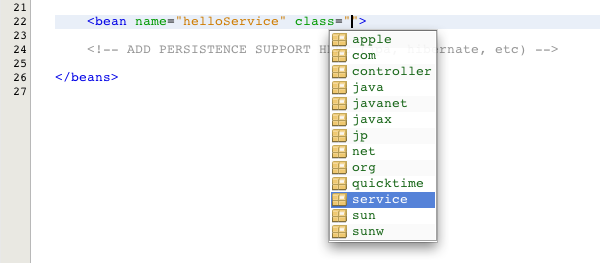
Por último, haga clic de nuevo en el Source Editor y seleccione las importaciones Fix. Nótese que la instrucción siguiente se añade a la parte superior del archivo:

importar service.HelloService;

Todos los errores deben fijarse ahora.

* Registrarse HelloService en applicationContext.xml. Abra applicationContext.xml en el editor de código y entrar en la declaración de frijol siguiente:

*<bean name="helloService" class="service.HelloService" />*

****